SYSTEM RESOURCE INFO TOOL

Display Size and Limits of C Data Types

Submitted By:

Nirnay Sinha

Under the Guidance of:

Miss Naina Devi

Subject:

Programming Fundamentals with C

College:

Rungta International Skills University

Agenda 📌

📘Project Introduction

🛠️ Tools & Technologies Used

💻System Resource Info Tool – Code

🧠Code Explanation

📊Program Output

🔍Output Explanation

🌍Applications

⭐ Advantages

🚀Future Scope

✔️ Conclusion

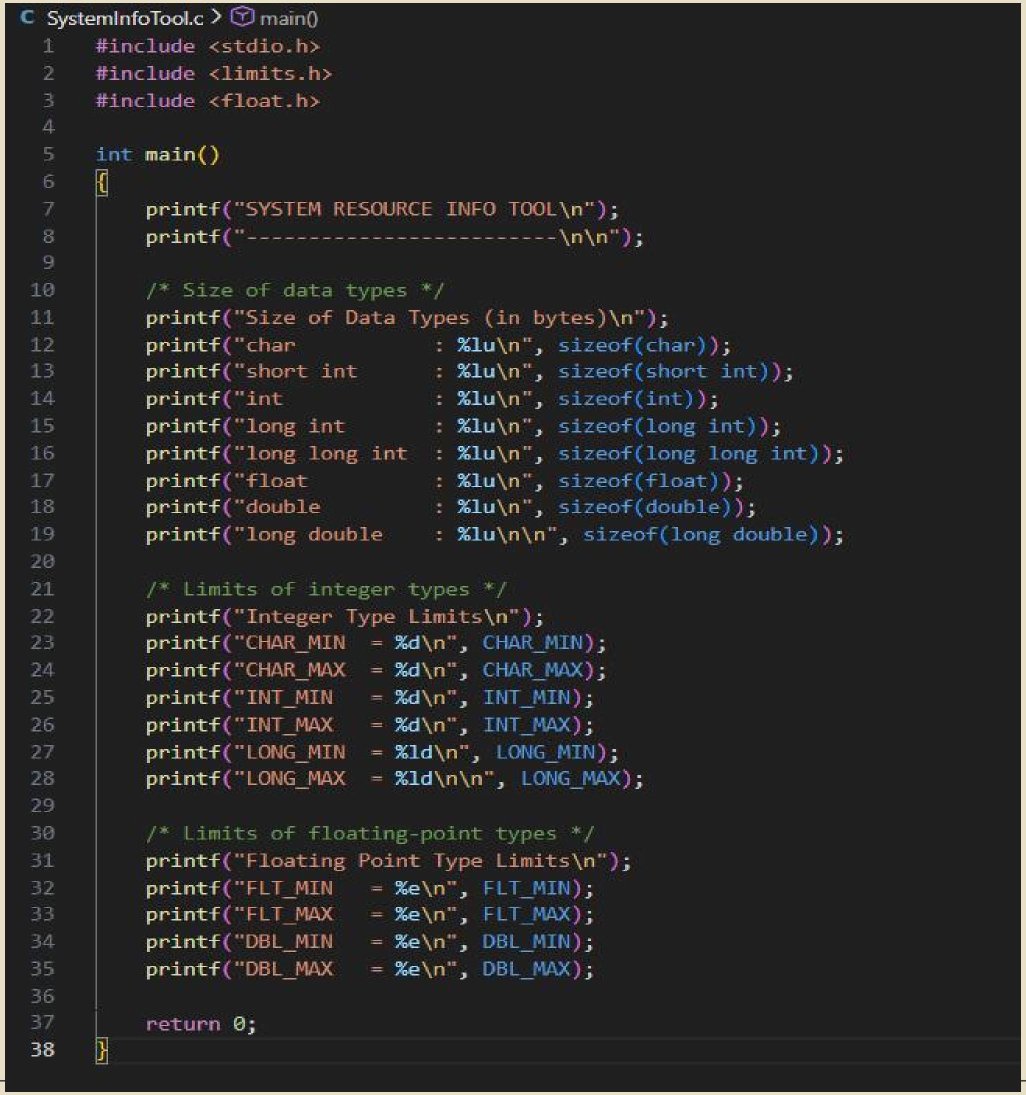
Project Introduction

- System Resource Info Tool is a C-based program.
- It displays:
  - Memory size of C data types
  - Their minimum and maximum limits
- Uses standard header files for portability.
- Helps understand system-dependent behavior of C programs.

Tools & Technologies Used

- Programming Language: C
- IDE: Visual Studio Code
- Compiler: GCC / MinGW
- Header Files Used:
  - `stdio.h`
  - `limits.h`
  - `float.h`
- Operating System: Windows / Linux

System Info Tool

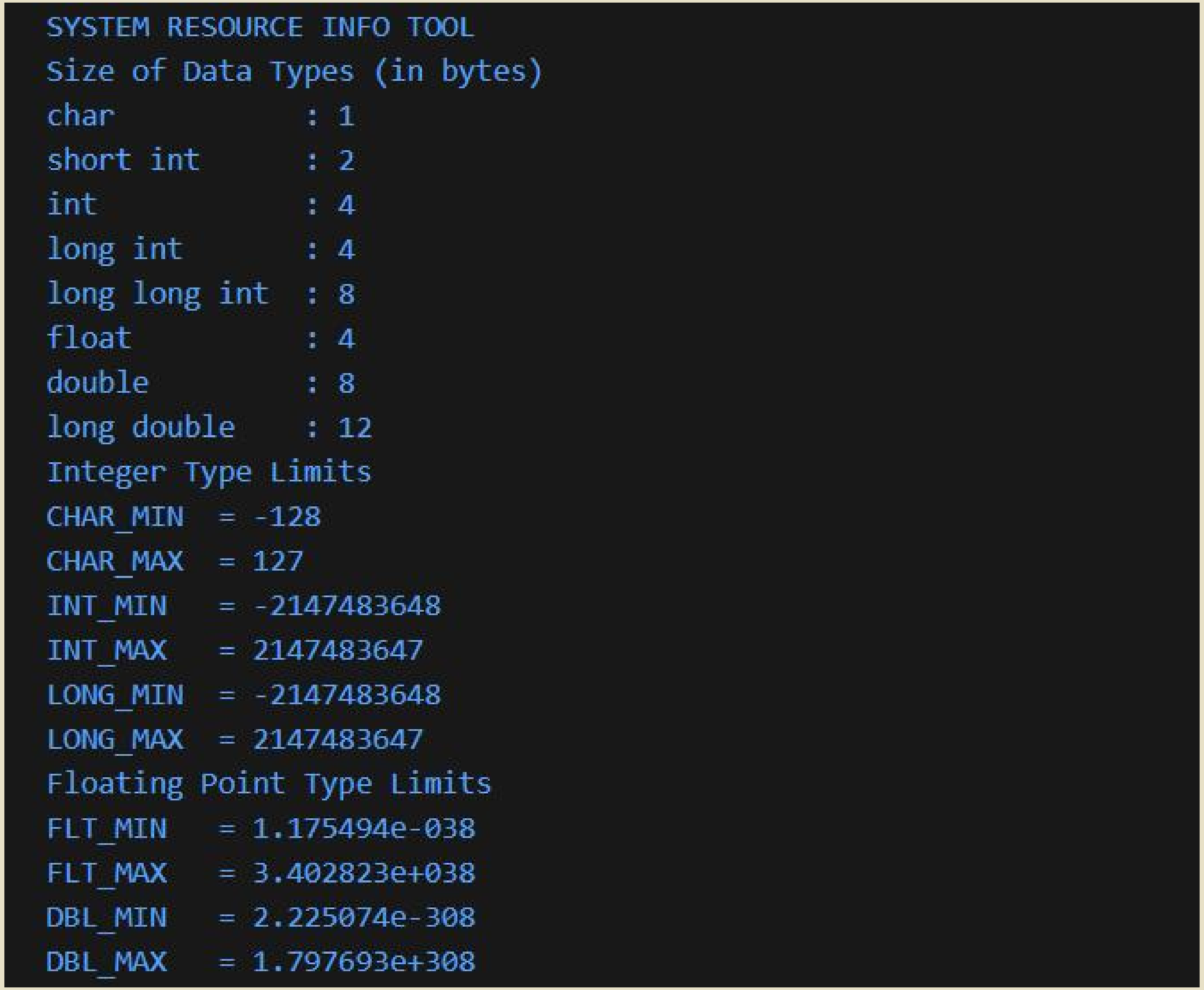


```c
#include <stdio.h>
#include <limits.h>
#include <float.h>

int main()
{
    printf("SYSTEM RESOURCE INFO TOOL\n");
    printf("-------------------------\n\n");

    /* Size of data types */
    printf("Size of Data Types (in bytes)\n");
    printf("char           : %lu\n", sizeof(char));
    printf("short int      : %lu\n", sizeof(short int));
    printf("int            : %lu\n", sizeof(int));
    printf("long int       : %lu\n", sizeof(long int));
    printf("long long int  : %lu\n", sizeof(long long int));
    printf("float          : %lu\n", sizeof(float));
    printf("double         : %lu\n", sizeof(double));
    printf("long double    : %lu\n\n", sizeof(long double));

    /* Limits of integer types */
    printf("Integer Type Limits\n");
    printf("CHAR_MIN  = %d\n", CHAR_MIN);
    printf("CHAR_MAX  = %d\n", CHAR_MAX);
    printf("INT_MIN   = %d\n", INT_MIN);
    printf("INT_MAX   = %d\n", INT_MAX);
    printf("LONG_MIN  = %ld\n", LONG_MIN);
    printf("LONG_MAX  = %ld\n\n", LONG_MAX);

    /* Limits of floating-point types */
    printf("Floating Point Type Limits\n");
    printf("FLT_MIN   = %e\n", FLT_MIN);
    printf("FLT_MAX   = %e\n", FLT_MAX);
    printf("DBL_MIN   = %e\n", DBL_MIN);
    printf("DBL_MAX   = %e\n", DBL_MAX);

    return 0;
}
```

Code Explanation

- The program starts by including standard header files:
- `stdio.h` for input and output operations
- `limits.h` to obtain minimum and maximum values of integer data types
- `float.h` to obtain range information of floating-point data types
- The `main()` function is the entry point of the program.
- The `sizeof()` operator is used to determine the **memory size (in bytes)** of various C data types such as:
- `char`, `int`, `long`, `float`, `double`, etc.
- Predefined macros like `INT_MAX`, `CHAR_MIN`, `FLT_MAX`, and `DBL_MAX` are used to display the **limits of data types**.
- `printf()` statements are used to format and display all information clearly on the screen.
- The program terminates successfully using `return 0`.

Output



```
SYSTEM RESOURCE INFO TOOL
Size of Data Types (in bytes)
char           : 1
short int      : 2
int            : 4
long int       : 4
long long int  : 8
float          : 4
double         : 8
long double    : 12
Integer Type Limits
CHAR_MIN  = -128
CHAR_MAX  = 127
INT_MIN   = -2147483648
INT_MAX   = 2147483647
LONG_MIN  = -2147483648
LONG_MAX  = 2147483647
Floating Point Type Limits
FLT_MIN   = 1.175494e-038
FLT_MAX   = 3.402823e+038
DBL_MIN   = 2.225074e-308
DBL_MAX   = 1.797693e+308
```

Output Explanation

- The output displays **system-specific information** about C data types.
- It shows the **memory size (in bytes)** consumed by each data type using the `sizeof()` operator.
- Integer limits such as **minimum and maximum values** are obtained from `limits.h`.
- Floating-point ranges and precision are obtained from `float.h`.
- The displayed values **may vary across different systems and compilers**, proving that C is system dependent.
- This output helps programmers **select appropriate data types**, ensuring safe, efficient, and portable programs.

Applications

- **Real-world Applications of System Resource Info Tool:**
- Ensures **portable and reliable C programs** across different systems
- Helps developers in **embedded systems** with limited memory resources
- Useful for **system-level and low-level programming** (drivers, OS modules)
- Assists in **memory optimization and management analysis**
- Serves as an **educational tool** for learning C data types and system behavior

Advantages

- **Key Benefits of the Tool:**
- **Simple and efficient** way to obtain system-specific data type info
- Helps **prevent overflow and underflow errors** in programs
- Relies on **standard C libraries** for accuracy and portability
- Enhances understanding of **how data types interact with system memory**
- **Flexible and easy to extend** for additional features in future

Future Scope

- **Potential Enhancements:**
- Add a **menu-driven interface** for interactive user selection
- Include **user-defined and complex data types** like structures and unions
- Compare system resource info across **multiple platforms and compilers**
- Export collected data to **text or CSV files** for analysis
- Develop a **GUI version** for easier visualization and professional use

Conclusion

- **Summary of the Project:**
- The **System Resource Info Tool** successfully displays the **size and limits** of all C data types
- Highlights **system-dependent behavior**, emphasizing portability and reliability
- Acts as a **reference for safe and optimized programming**
- Fulfills the project objectives of educating users about **memory usage and data type constraints**

😊 Thank You!

Any Questions?